The MMSFORTH Newsletter

MILLER MICROCOMPUTER SERVICES
61 LAKE SHORE ROAD, NATICK, MASS. 01760
(617) 653-6136

Vol.2 No.5

MAKING UP FOR LOST TIME (Editorial)

So, we're a little late. What's that? Okay, then, a lot late! Here's what happened...

As many of you know, The MMSFORTH Newsletter was running behind schedule last Fall, as MMS was beginning major new work on the IBM Personal Computer and also continuing projects on General Ledger, FORTHWRITE and more. Under the pressure of it all, we didn't deliver the "September-October 1981" issue until early March 1982. Things didn't relax, so here's the next issue in September 1982. At least for a while -- and for all too obvious reasons -- we'll just number the masthead instead of dating it. The dates may have been misleading, but you **will** receive the pre-planned number of issues. In fact, we'll add one more than promised in Volume 2, to give it 6 issues like the other ones.

MMS is humming along in other departments. The IBM PC/8088 version of MMSFORTH is in very active distribution, very good, and keeping us busy producing copies and designing extra features. We've introduced an advanced Graphics Utility for it this month, and we're beginning to believe that the IBM PC can support **unlimited** extra features. FORTHWRITE, producing this issue in full proportional type for the first time, has been solving user problems as a draft release for nearly a year and is about to go into general distribution. Our General Ledger software is also solving client problems now, and completion of other projects permits the writing of its manual, like this Newsletter, to get back into action.

We've been getting about, as well. Exhibiting at shows in Boston last October and New York City in April, two Forth conferences in May in Washington, D.C. and Rochester, N.Y. Coming up: an early October California visit for another FORML Conference at Asilomar and the Forth Convention in San Jose, two talks in one night at the New England Computer Society's November 3rd meeting, then that big, big Boston show again from November 11th through 14th. We've completed some impressive custom projects for clients, as well. And we've been teaching Forth on a client basis. We're about to offer some Forth courses to our general users, as well.

Our friends have been busy, too. They have been publishing articles on MMSFORTH and related items in 80-Microcomputing, BYTE, Desktop Computing, and other magazines -- each of these authors receives money from the publisher, of course, and MMS thanks them with another MMSFORTH software product, too! You've also been submitting **lots** of interesting articles which we again can share via this Newsletter -- and each one printed earns a year's free Newsletter renewal, again as our way of saying thank you on behalf of all the other readers.

-- A. Richard Miller, Editor 4th Class

NEW AT MMS

THREE FEATURED BOOKS:

INTRODUCTION TO FORTH, by Ken Knecht	$9.95
SYSTEM GUIDE TO FIGFORTH, by Dr. C.H. Ting	$25.00
INCREDIBLE SECRET MONEY MACHINE, by Don Lancaster	$7.95

Our new books for this issue start with Ken Knecht's beginners' book on our own favorite version of Forth, MMSFORTH. Well, not quite our favorite version; this 1982 book introduces MMSFORTH V1.9 which was superseded by V2.0 in June 1981. The book is not at all advanced, and some readers feel it is written without enough deep understanding of the Forth environment. On the plus side, this paperback is well printed, costs only $9.95 plus S/H, discusses MMSFORTH words in considerable detail, and it does so from the perspective of one who is familiar with Radio Shack Level II BASIC - a good starting point for many new users. Don't expect it to compete with STARTING FORTH, which remains our favorite. Do consider adding it to your Forth library for another view, particularly for a beginning, BASIC-oriented programmer. We are pleased to announce this first "outside" book about MMSFORTH, and would encourage a second edition which talks about our current version.

THE SYSTEM GUIDE TO FIGFORTH is another "maybe" book for our Forth programmers. At $25.00, it will be of maximum interest to those who run figFORTH rather than our own MMSFORTH. However, since it describes and analyzes much of the inner workings of a Forth system, advanced Forth programmers who found THREADED INTERPRETIVE LANGUAGES interesting and helpful may value it as a practical look inside a simple and popular Forth system.

Our third "book of the month selection" is not even about computers. But if you or a friend have contemplated starting your own business or have taken the plunge, we strongly recommend this witty and insightful how-to (or why not to!) book on the many facets of personal-sized entrepreneurial ventures. This book is right-on and its information is worth much more than its small price of $7.95. Hint: many readers come back for additional copies, because it is an unusual and thoughtful gift which will interest many of your friends, too.

What? Why yes, of **course** they are in stock at MMS!

MMSFORTH USER WORKSHOP

Here's the workshop many of you have been asking for! It's an intensive two-day MMSFORTH workshop on Friday and Saturday, October 29th and 30th in Lowell, Massachusetts (near Boston). Cost is $225 per attendee, including lunches and snacks but not including transportation and lodging. Because it will be a hands-on session, a few $50 discounts are available to attendees who bring their computers for group use.

Principal instructor is our own Tom Dowling, primary author of the MMSFORTH System, with able assistance from Dick Miller and others. Topics will include an MMSFORTH overview, programming in MMSFORTH, and special programming tips on our popular applications: THE DATAHANDLER, FORTHCOM and FORTHWRITE. Within the time constraints, we will attempt to be responsive to specific audience interests as well. Preparation on your part will consist of reading STARTING FORTH. (If reading it left some questions, bring them too!)

To attend, send your check to MMS now for $225. Maximum attendance is limited to 20 persons; if the workshop is full or if we fail to get a minimum of 10, all your money will be refunded. If you cancel, your prepayment will not be refunded unless we get a replacement attendee. If you can't make this workshop, please tell us if you'd like to attend at another time.

INSIDE MILLER MICROCOMPUTER SERVICES

FORTHWRITE

The design and programming of FORTHWRITE, our hot-shot word processor, are complete and final documentation is expected by late October. FORTHWRITE now has the ability to accept DATAHANDLER files and to treat their record fields as text blocks -- inserting them as is, or in various special formats -- to create one personalized copy of your document for each DATAHANDLER record. This makes it one of the most powerful **form letter systems** available, and will suggest many other applications to our users.

FORTHWRITE also offers **executive quality printing.** Full proportional control is directly available for letter quality printers including Radio Shack's Daisywheel II, most NEC Spinwriter models, the NEC 8023A-C and the C.Itoh Prowriter 8510. Most other letter quality printers can be set up -- by you or, on a consulting basis, by MMS -- by editing their control codes into tables provided in our printer driver source code. And when we say full proportional, we mean much more than most other word processors can deliver: right justification, tabbed columns (not easy with varying character widths!), and the ability to intermix proportional and various constant pitches.

Harold Baughan of Alpine Systems has assisted us in creating the FORTHWRITE manual. He's made it quite complete, yet fun to read and to learn. For $175.00, you can add FORTHWRITE to your MMSFORTH System. We think you'll be mighty glad you did!

GENERAL LEDGER

Beta test sites are running well, but final documentation is still underway. Watch this space for further announcements.

TURTLE GRAPHICS FOR IBM PERSONAL COMPUTER

Now that the IBM PC version of MMSFORTH is equipped with all our normal goodies, we're forging ahead with additional capabilities for this advanced machine. First to come available is TGRAPH, a new graphics utility with the raw materials for fast and sophisticated graphics implementations. TGRAPH includes one such implementation which should please many users: a complete set of Turtle Graphics, the popular LOGO language version of vector graphics which is so easy to use and to explain. As we expected, what is easy in LOGO is even more natural to Forth. Build your own graphics words, while drawing diagonal lines at the rate of 100 lines per second!

The MMSFORTH Graphics Utility requires the IBM Color Adapter (that's IBM's name for an accessory board) and will support color or monochrome displays. It is included on the IBM PC Utilities Diskette at no extra cost. Earlier IBM PC Utilities Diskettes can be rewritten for the usual $10 plus $2 shipping/handling.

SOME FORTH UTILITY WORDS
by Russ Hensel

The self-extensible nature of Forth encourages users to develop words that are useful in as many different programs as possible. Some of the words do not work out as well as was planned, but over time most programmers arrive at a collection of words that they find particularly useful. In the year and a half that I have been using MMSFORTH I have developed a few such words that I think other programmers would also find useful.

DISPLAY COMMENT LINES DURING LOADING

The first item I would like to share is a procedure, not a word. Many programmers put a comment at the beginning of each block to label it. This is especially useful in conjunction with the word INDEX, which will list the first line of a range of blocks. I have adopted a variation of this procedure; I use a dot-quote to print my comment on the screen as the block is compiled. The two principal reasons I prefer this to a comment are:

* It makes the machine look busy as it compiles.
* It can have debugging advantages; for example, if a dropped --> causes compilation to end earlier than planned.

I have standardized the string so that it contains the block number being compiled, the name of the set of words, the names of the most important words on the block, and finally the version number and date. (Ed: MMS puts the date on the left, instead!) For examples of this procedure in use see the blocks of code includes later in this article.

REPORT WHAT EXTENSIONS ARE LOADED

Most programmers build up sets of words that are used as building blocks for later programs. The set that is used to work with strings is an example that comes with the basic MMSFORTH disk. Afer compilation, it is not clear which sets are present. Of course you can use the word FIND to test for the presence of a word, but there is an easier way. My first approach was to use a dot-quote in conjunction with the common procedure of using TASK to mark the beginning of a set of words. The result was

```
: TASK CR ." STRINGS 011282 " ;
```

When you type TASK the computer prints the name of the TASK that has been compiled.

The problem with this approach is that each use of TASK hides the earlier uses. What I wanted was to see a list of **all** of the tasks which had been compiled. A bit of thought gave me a new solution. The definition of TASK is changed just a bit more, so it includes TASK, a reference to earlier TASKs:

```
: TASK  TASK CR ." EXAMPLE" ;
```

OUTPUT-FORMATTING WORDS

Formatting of output is often a pain in the neck. I have found the following two words to be helpful.

TAB This word works much like the Basic TAB function. n TAB will move the cursor to position n across the screen. If it is already past position n, the cursor is not moved.

1/2CR This word does half a carriage return. You might ask what half a carriage return is. If the cursor is more than half way across the screen 1/2CR is the same as a carriage return; otherwise it moves the cursor to the halfway point. 1/2CR gives the programmer a very simple method of setting up 2 columns down the page. Extensions to produce words such as 1/2CR are straight-forward to write.

Here is the code for these words. Note that I have used the techniques discussed earlier in coding up the block.

```
 0 CR ." 57  TAB                                        2.0-121281 "
 1 : TASK TASK CR ." TAB 121281" ;
 2 : TPOS  ( -> n ) ( Find position across video screen = n )
 3       16416 @ 15360 - 64 MOD ;
 4
 5 : TAB TPOS - 0 MAX SPACES ; ( n -> ) ( Tab to position n )
 6
 7 : 1/2CR    ( -> ) ( Do half a cariage return )
 8         TPOS 32 < IF 32 TAB ELSE CR THEN ;
 9
10 (       --> ( )
11 (  : TEST PAGE      20 0 DO I . 1/2CR LOOP KEY ; )
```

SOME TOOLKIT WORDS

Like MMS, I have a few words that I consider to be tool kit words. Some of them are especially useful to users that have only one disk; others are more generally useful.

C-D This word is used when you need to change disks. It insures that data is not transferred between disks or erronously read when disks are swapped.

T This is a simple word useful for moving blocks from one disk location to another. You can have the blocks on the same disk or on different disks. If any updated blocks are in memory, use C-D or FLUSH first. T does an EDIT and FLUSH .

MEM and .MEM This is just a rearrangement of the MMSFORTH words. I often like to have printing and non-printing versions of words.

0DMEM, DMEM and .DMEM These three words are used together to measure the amount of memory used in compiling other words. The procedure is as follows:

a) Set the count of used memory to 0 with 0DMEM. You can check that it is 0 with .DMEM.
b) Compile whatever you want.
c) Print the amount of memory used with .DMEM. The count is not set back to 0 until 0DMEM is used again. DMEM is just a nonprinting version of .DMEM.

The words in the following two blocks are built upon the TOOLKIT Extension in MMSFORTH:

```
 0 CR ." 60  RH TOOLKIT                                 2.0-112081"
 1 CR ." USES MMS   06/28/81 TOOLKIT: .MEM  .S  TRY  RUN  2EDIT  "
 2 FORTH DEFINITIONS
 3 : TASK TASK CR ." RH T K 011582" ;
 4 : .S ( Non-destructive stack print-out )
 5   DEPTH 0> IF S0 @ DEPTH 1 DO 2- DUP @ . LOOP DROP THEN ;
 6 : NOT-FOUND ."  Word not in dictionary" ;
 7 : TRY ( TRY name   show stack before & after )
 8 CR .S ." -> " FIND ?DUP IF EXECUTE .S ELSE NOT-FOUND THEN CR ;
 9 : RUN (   load screen just edited )        SCR @ LOAD ;
10
11 ( : 2EDIT ( drive0-block# -> , drive 0:1 comparisons )
12 ( FLUSH EMPTY-BUFFERS DUP :1 BLOCK DROP EDIT ;  )
13 ( : 2EDITS ( drv0-blk# #blks ->    OVER + SWAP DO I 2EDIT LOOP ;
14      )          -->
```

```
 0 CR ." 61  RH TOOLKIT                                 2.0-112081"
 1 : C-D* ( -> )       FLUSH EMPTY-BUFFERS ;
 2 : C-D  ( -> )       C-D* CR ." DISK CHANGE OK" ;
 3 : T    ( n1 n2 -> )  (  Copy from bk n1 to bk n2 )
 4     DUP ROT SWAP EMPTY-BUFFERS EDITOR COPY
 5     FORTH  CR ." INSERT DEST. DISK" ENTER  EDIT FLUSH        ;
 6 : MEM  ( -> N )  (  N= available bytes of RAM )  'S PAD -    ;
 7 : .MEM ( -> )    ( Print MEM                  )   MEM U.     ;
 8 0 CONSTANT   0MEM   ( Used to store amount of free memory )
 9 : 0DMEM ( -> )  ( Reset 0MEM )    MEM ' 0MEM !   ;
10 : DMEM  ( ->N ) ( n = delta mem ) 0MEM MEM - 2 - ;
11 : .DMEM ( -> )  ( prints DMEM )           DMEM . ;
```

MODIFIED **VAL**

I often like to work with strings and then convert them to numeric values. I discovered that VAL is not always the most satisfactory way of doing this. The most important difficulty is that VAL returns 0 if it encounters a leading + sign or a blank. I have written a new version, VAL+, that is listed on the following block. Also listed is a double precision version, DVAL. DVAL must be included since it is used in the definiton of VAL+.

```
 0 CR ." 63   DVAL  VAL+                                2.0-011582 "
 1 : TASK TASK CR ." DVAL 011582" ;
 2
 3 ( These words  can deal with leading blanks or + sign )
 4
 5 : DVAL   ( $ -> D ) ( Convert string to DP number    )
 6       HERE $! 0  HERE HERE C@ 1+ +     C! HERE
 7       BEGIN DUP 1+   C@ DUP   32 = SWAP 43 = OR
 8             IF 1+ 0 ELSE 1 THEN
 9       UNTIL   NUMBER DROP #PT C@    IF ELSE   HI#  @  THEN ;
10 : VAL+    ( $ -> N )             DVAL DROP ;
11
12 (  : TEST  BEGIN  99 IN$ VAL+  CR  . SPACE .  0 UNTIL   ; (  )
```

MMSFORTH MODIFICATIONS

MOD FOR IBM PC SYSTEM DISK

Are you an IBM PC user? Please make these two mandatory changes if your System Diskette was produced prior to August 26th, 1982.

Edit Block 53, Line 8 to read as follows:
OR IF DROP -1 LEAVE THEN LOOP DUP -1 <>

Edit Block 48 by shifting a blank line to Line 8, then extending Line 7 onto Line 8 to add GTC PTC just before the ;.

Then, reprecompile the new changes aboard by editing Block 20, Alternate-S, entering the items listed in its Line 1, and answering the CUSTOMIZE queries in your usual manner when prompted.

Also, when you are ready to use Turtle Graphics you will want to know that the IBM PC System Disk's Block 50 has been modified to add graphics words for HIRES and MDRES. You will COPY it across from Block 50 of the new V2.0 IBM Utilities Disk -- the one with Turtle Graphics.

MOD FOR IBM PC PRINTER CABLE

Here's an easy fix to speed up Spinwriters and many other printers when used with the IBM PC. The IBM PC printer cable can be modified to greatly speed throughput by unsoldering the wire from Pin 10 (-Acknowledge) at the printer end, and resoldering it onto the one already on Pin 11 (+Busy).

If you can't stop to make that fix, here's a simple expedient for FORTHWRITE users: just use the Front Menu's Defaults Utility or FORTHPRINT's "Set new printer-driver" Option to reduce the printer buffer size to 2 characters.

DATAHANDLER CUSTOM MODIFICATION TECHNIQUES

MOD TO DATAHANDLER V1.2

When using THE DATAHANDLER, we recommend against trying to cram in the last record or two. Sooner or later, some necessary addition won't fit. But occasionally too much gets put in anyway, so in the prior issue we gave some optional ways to guard against space-overflow problems with THE DATAHANDLER.

This month, MMS thanks Peter Wilkins of New Hampshire and Michael Cook of England for reporting another space-overflow problem - stemming from filling a file close to the limit and then expanding it further with the CHANGE command. The fix is easy, and uses so little RAM that MMS has made it a standard modification. Modify the CHANGE definition in your copy of THE DATAHANDLER by inserting a few words into Lines 4 and 5 of Block 40, as follows:

```
4 I CHANGE-IT I LRC ! IF I -RECORD +RECORD ?DUP IF I IRECORD
5 ELSE I -INDEX $C# THEN ELSE LEAVE THEN ELSE DROP THEN LOOP
```

That's the whole change, other than reprecompiling your DATAHANDLER diskette. Now, should you CHANGE the final record and make it oversize, instead of harming the files directory it will just lose the last record and abort the operation. If you don't have a printout of the last record, you can reload the file as it existed at the prior SAVE operation and proceed again with greater care.

INSIDE TRACK (for intermediate users)

AN IMPROVED WAY TO USE CUSTOMIZE

Last issue's article on customizing a NOTEPAD diskette provided a method for dropping the DIRectory, loading (compiling) new source code into memory, and then using MMSFORTH's CUSTOMIZE Utility to precompile the whole shooting match back onto disk for direct start-up. A good idea, but now we present a better way to load CUSTOMIZE once you no longer have the DIRectory aboard - just put it back aboard! The new method goes like this:

a) Enter: **FORGET DIR EMPTY-BUFFERS**
b) Load appropriate blocks of source code.
c) Enter: **DIR CUSTOMIZE**

This way, there is no need to know the starting block number of the CUSTOMIZE routine. The basic scheme is to strip out the directory from under your new words, then to temporarily reload the DIRECTORY words for two reasons: so Forth knows the block numbers for CUSTOMIZE, and because CUSTOMIZE expects one DIRECTORY word, ->, to be available when it does its thing.

Why the **EMPTY-BUFFERS**? Because if, just if, your new routine were to be loaded from one disk on the same block numbers as you had just loaded from another, without **EMPTY-BUFFERS** the last two same-numbered blocks would load from memory instead of from the new diskette.

ADD BOLD-PRINTING TO THE NOTEPAD

While we're on the subject of last issue's CUSTOMIZE article, here's a nice additional touch for those whose printer can bold-print. Add two words to your precompiled NOTEPAD program, YBOLD and NBOLD ("yes-bold" and "no-bold"), to toggle the printout to and from the bold, or enhanced, typeface. In this example, we will program for the popular Epson MX-80 printer which recognizes the decimal code sequence 27, 69 to begin bold printing and recognizes the sequence 27, 70 to return to normal print. Simply add the following to the definitions in the NOTEPAD source code, then reprecompile:

```
: YBOLD   PRINT  27 EMIT  69 EMIT  CRT ;
: NBOLD   PRINT  27 EMIT  70 EMIT  CRT ;
```

Now, you can print draft versions of a NOTEPAD letter or other text in normal characters by Entering shift-control-P. Once it looks the way you wish, reset the printer to bold-print (which is correspondence quality, but twice as slow) by pressing shift-control-Q to quit, Entering YBOLD to change font, Entering NP to return to the same text, and pressing shift-control-P again to reprint in the new, bold typeface.

CUSTOMIZING MMSFORTH FOR SPECIAL DISK DRIVES

MMS receives many requests concerning the modification of MMSFORTH to support special disk drive configurations. In general, all combinations are possible and many can be done with little or no programming. However, MMS has not done them all and does not offer custom hardware interfacing except on a consulting basis. To help you decide whether to go it alone, here is further information on the subject.

Our CUSTOMIZE Utility offers the simplest level of modification for MMSFORTH and is sufficient for most needs. It enables you to set your system diskette for your actual hardware: from 1 to 8 disk drives, each of which may have up to 255 tracks of single or double density. It also sets for Drive 0 start-up speed, and permits setting the track to track stepping speed individually for each drive (TRS-80) or for all at once (IBM PC).

CUSTOMIZE presumes that on TRS-80's single density (SDEN) means 10 256-byte sectors per track and that double density (DDEN) means 18 256-byte sectors per track. On the IBM, its own format (IBM) means 8 512-byte sectors per track and M.3 is equivalent to the DDEN format, above. However, these normal settings may be further adjusted by direct modification of the numbers within MMSFORTH's DISKDATA array. For example, let's say you had one 40-track, single-sided drive (Drive 0) and now have just purchased an 80-track, double-sided disk drive. Like most, it was set up in hardware to look like two separate drives, so let's use the standard CUSTOMIZE Utility to define it as if it were those two separate 80-track drives: Drive 1 and Drive 2. Because CUSTOMIZE automatically sets to the "front" side of each drive, we must then manually reset "Drive 2" to the "flip" side of Drive 1, and finally we will use a modified version of CUSTOMIZE to save this modified system to disk without undoing the new values.

Let's look more closely. The new disk drive was set up as two 80-track drives with 18 sectors per track (i.e., with 80 tracks on each side of the diskette), so CUSTOMIZE it above the single 40-track drive, as two single 80-track drives. Since it is to become your second drive (Drive 1), examine its sectors/track setting in your MMSFORTH System. We can observe the current DISKDATA settings with 0 DISKDATA 50 DUMP, but to break the array into a more readable display we prefer to dash off a temporary Forth word, **SHOW-DISKDATA,** at this point:

```
: SHOW-DISKDATA    CR  0 DISKDATA  4 0
      DO  DUP 9 ( 9 for TRS-80, 10 for IBM ) + DUP ROT
          DO  I C@  4 .R
          LOOP  CR
      LOOP  DROP ;
```

Using this new tool, we observed that the Model III settings following the CUSTOMIZE operation read:

```
SHOW-DISKDATA
 179   0   0   1   0  39   1  19 129
 103   1   0   1   0  79   1  19 130
 103   1   0   1   0  79   1  19 131
   0   0   0   0   0   0   0   0   0
```

With the aid of the DISKDATA table in Appendix 12 of your MMSFORTH USERS MANUAL, observe that the elements of the DISKDATA array now displayed consists of 9-byte (TRS-80) or 10-byte (IBM) data strings, one describing each "Drive". The first two columns, or bytes, give the number of 1024-byte Forth blocks on that "disk". Each byte can only count to 255, but the computer reads the two bytes together as 103 + 1*256 = 359. 2 DISKDATA puts the starting address of the "Drive 2" string on stack; therefore 2 DISKDATA 8 + is the element which defines the drive number as well as the front or back surface for this "drive". Our trick is to change this element's value from the "front" side of "Drive 2" to the "flip" side of Drive 1. According to the information in Appendix 12, this is done by ANDing a 16 to the value for Drive 1 (16 + 130, or 146). We tuck it into place with a 146 2 DISKDATA 8 + C!.

Now use SHOW-DISKDATA again to verify the new settings:

```
SHOW-DISKDATA
 179   0   0   1   0  39   1  19 129
 103   1   0   1   0  79   1  19 130
 103   1   0   1   0  79   1  19 146
   0   0   0   0   0   0   0   0   0
```

Then use a **modified** CUSTOMIZE - one which **won't** reset the disk drive parameters again - to save the new system back onto the diskette. Make the modification by editing a pair of parentheses around the DISK? definition in Block 3 of the CUSTOMIZE Utility. Then use this special CUSTOMIZE to move our new DISKDATA array back from memory to disk. Voila, we're done!

The clues for many more configurations are given in the DISKDATA section of the System Constants Tables, in Appendix 12 of your MMSFORTH USERS MANUAL. For example, here is how an equivalent IBM DISKDATA array looks when set-up for two 40-track, double-sided drives:

```
SHOW-DISKDATA ( re-defined for 10 elements )
 159   0   2  42   0  39   1   9 128   2
 159   0   2  42   0  39   1   9 144   2
 159   0   2  42   0  39   1   9 129   2
 159   0   2  42   0  39   1   9 145   2
```

GET-TOGETHER

Share your questions and answers with a MMSFORTH User Group! Contact MMS for help to start one in your region, or to revive one which seems inactive. Here is our present list of contacts for local MMSFORTH User Groups:

CA: Earl Mortensen, 974 Pleasant Hill Road, Redwood City 94061 (415/367-9882).
CA: Ken Nonomura, 416 Duncan Street, Apt. 5, San Francisco 94131 (415/285-5062).
CA: Morris Herman, 503 Rosario Drive, Santa Barbara 93110 (805/964-7144).
CA: Rich Royea, 6456 Lubau, Woodland Hills 91367 (213/704-6859).
FL: Bob Vest, 64 NW 111th Street, Miami Shores 33168 (305/751-7511 eves.).
IL: Walter Cooper, 5112 West 30th Place, Cicero 60650 (312/656-6183).
LA: Ed Laughery, 1222 Jason Drive, Denham Springs 70726 (504/665-7537).
MA: Jim Gerow, 22 Crestwood Drive, Framingham 01701 (617/443-9521 x3599 days, 872-1882 eves.).
MI: Bob Zwemer, 6408 South Washington, Lansing 48910 (517/393-9287).
NB: Bill Schneider, 1425 North 14th Street, Lincoln 68508 (402/786-2715 days, 476-3671 eves.).
NH: Gregg Reed, RR2, Box 167, Dumbarton 03301 (603/774-5311 eves.).
NY: Bernie Bergman, c/o Eagle Jewelry, 201 Canal Street, New York City 10013 (212/966-3414 days, 201/339-5575 eves.).
PA: Gus Raab, 806 Freedom Circle, Harleysville 19438 (215/368-4866)
TN: Cliff Fiedler, 1908 Belcourt Avenue, Nashville 37212 (615/327-9123).
TX: Larry Goforth, 10203-J Golden Meadow, Austin 78758 (512/836-0981).
TX: Jim Shepard, 16210 Arbor Downs Drive, Dallas 75248 (214/661-9702).
TX: Dan Healy, 11511 Katy Freeway, Suite 150, Houston 77079 (713/496-4660 days).
WA: Rod Proctor, 13520 N.E. 29th Place, Bellevue 98005 (206/453-0635 days, 883-1923 eves., and MicroNet 70110,402).
AUSTRALIA: Peter Wragg, 2 Jilba Street, Indooroopilly, Queensland 4068 (07/378-1623, and CL1641 on The Source).
AUSTRALIA: Dave Dartnall, 20 Eldon Street, Dianella, Western Australia 6062 (09/446-8100).
CANADA: Kalman Fejes, 1149D Meadowlands Drive East, Ottawa, Ontario K2E 6J5 (613/225-2443).
ENGLAND: John Newgas, 1 Philip Court, 89 Hornsey Lane, Highgate, London N6 5LN (01/539-7071 days, 348-6518 eves.).
JAPAN: Akira Akutsu, M.D., 2-176 Issha, Meito, Nagoya, 465.
WEST GERMANY: Nigel Head, Birngartenweg 93, 6100-Darmstadt

NOTE: Program trading is one popular facet of these meetings, but NOT commercial programs and WITHOUT MMSFORTH SYSTEMS aboard! Promote legitimate sharing, discourage pirating, and take care not to jeopardize your own MMSFORTH serial number.

CRYPTOQUOTE PUZZLE, by Jill Miller

The solution to last issue's cryptoquote was:
"THE FIRST THREE WORDPROCESSORS WERE SO-SO,
BUT WE WROTE THE FORTHWRITE! - MMS"

Here is a new cryptoquote to solve using the MMSFORTH GAMES DISKETTE. It's from an old folksong and could be called the Forth programmer's lament:

QTMLM EHO H EBLX J FJDTQ THCM OHJX,
WRQ ETHQ JQ EHO J THLXAN PGME.

PERIPHERAL TALK

TRS-80 SCREEN-PRINTER FOR PROWRITER AND NEC 8023 PRINTERS
(by Albert Strauss, with code edited by John Rible)

If you use one of the new C.Itoh Prowriter or NEC 8023 dot matrix printers, you know you've got a winner. But unlike the earlier Epson MX series, the Okidata 80, 82 and 83, and many others, these new printers don't support the TRS-80 graphics character set.

Al Strauss has trained these printers to dump your whole screen -- graphics characters included -- with the following three-block utility. As mentioned in prior Newsletter articles for other printers, we recommend setting to the printer's narrowest character pitch to yield screen-like proportions.

Block 4:

```
 0 ( 03/11/82 - ALPHA/GRAPHIC SCREEN PRINT, 1 of 2 )   : TASK ;
 1 ( Replaces SCREENPRINT for C.Itoh 8510 and NEC 8023 printers )
 2 ( Submitted by: Albert Strauss, Ridgewood N.J. )
 3 ( and Edited by: John Rible, MMS )
 4
 5 ( Note: EMIT type-style codes before using; )
 6 (       compressed-style gives truest screen reproduction )
 7 15360 CONSTANT SCREEN
 8 CREATE TABLE 0 C, 0 C, 15 C, 0 C, 0 C, 15 C, 15 C, 15 C,
 9 240 C, 0 C, 255 C, 0 C, 240 C, 15 C, 255 C, 15 C, 0 C, 240 C,
10 15 C, 240 C, 0 C, 255 C, 15 C, 255 C, 240 C, 240 C,
11 255 C, 240 C, 240 C, 255 C, 255 C, 255 C,
12 : SEND ( c -> ) BEGIN 14312 C@ 240 AND 48 = UNTIL 14312 C! ;
13 : 4TIMES ( c -> )  4 0 DO DUP SEND LOOP DROP ;
14 : SET1/3LF    27 EMIT ." T08" ; : SET2/3LF 27 EMIT ." T16" ;
15 : SETLF       27 EMIT ." T24" ;                              -->
```

Block 5:

```
 0 ( 03/11/82 - ALPHA/GRAPHIC SCREEN PRINT, 2 of 2 )
 1
 2 : SEND-SYMBOL ( index -> ) TABLE + DUP 1+ C@ SWAP C@
 3                27 EMIT ." S0008" 4TIMES 4TIMES  ;
 4 : LOW-SYMBOL  ( c -> ) 96 AND 2/ 2/ 2/ SEND-SYMBOL ;
 5 : HIGH-SYMBOL ( c -> ) 15 AND 2* SEND-SYMBOL ;
 6 : SECOND-PASS ( addr -> addr ) SET1/3LF 64 0 DO DUP I + C@ DUP
 7      128 <  IF DROP 32 EMIT  ELSE LOW-SYMBOL  THEN   LOOP CR ;
 8
 9 : PRINT-LINE ( addr -> ) SET2/3LF 64 0 DO DUP I + C@ DUP
10      128 <  IF 32 MAX EMIT  ELSE HIGH-SYMBOL  THEN  LOOP CR ;
11 HERE ]          PRINT SCREEN  16 0
12                 DO PRINT-LINE  SECOND-PASS  64 +
13                 LOOP SETLF CRT %CONT [
14 21 MMS 20 + !
15
```

MMSFORTH QUICKIES

'ARRAY' SOLUTION

Here is a solution to the 'ARRAY' problem in the prior issue of this Newsletter:

```
: LOOK   CR  6 SPACES  10 0
  DO  I 4 MOD  NOT
     IF  2 SPACES
     THEN  I .  SPACE
  LOOP  FLAGS 10 DUMP ;
```

THE LAST WORD: "In Scotland, the best programmers come from the Firth of FORTH!"
- Dr. Doug Margolis, Ossining, New York